Optimización de cadenas de adición

Fernando Aquino[1] y Guillermo Leguizamón[2]

[1] Universidad Tecnológica Nacional - Facultad Regional Concepción del Uruguay
fernando.aquino@hotmail.com.ar
[2] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
Universidad Nacional de San Luis
legui@unsl.edu.ar

Resumen El presente estudio aborda el problema de la computación óptima de cadenas de adición, ampliamente tratado con diferentes métodos y enfoques (tanto deterministas como estocásticos) y de interés en el ámbito de la criptografía. En este trabajo, se propone el uso del algoritmo de lobos grises o GWO (por sus siglas en inglés: Grey Wolf Optimizer) para hacer frente a este problema a fin de comparar los resultados obtenidos con otros enfoques del estado del arte. Si bien el problema en cuestión ha sido tratado mediante diferentes estrategias y para distintos tipos de exponentes, particularmente esta propuesta se centra en la optimización de cadenas de adición, asociadas a exponentes de tamaño moderado.

1. Introducción

Los algoritmos asimétricos de criptografía utilizan la exponenciación modular para cifrar y descifrar datos. El costo computacional producto de la gran cantidad de operaciones de multiplicación, que implica resolver potencias de número de orden considerable, se puede solventar mediante el uso del concepto de "cadenas de adición" [2]. Sin embargo, un exponente no posee una única cadena de adición válida asociada y por tanto hallar una cadena válida y cuya longitud resulte mínima constituye un problema NP completo. Se denomina exponenciación modular [8] a la operación cuya finalidad es obtener un valor b (entero, positivo) para satisfacer la siguiente ecuación:

$$b = A\,e \mod P \tag{1}$$

Dado un valor de A entero, arbitrario en el rango $[1, P-1]$ y un número e entero, positivo y arbitrario, se define exponenciación modular al problema de encontrar un valor único y entero positivo $b \in [1, P-1]$, capaz de satisfacer la ecuación (1), siendo para el problema estudiado aquí, la función objetivo a optimizar para poder obtener un valor b mínimo (i.e., una cadena de adición de longitud mínima.) El problema radica en que a medida que se incrementa el exponente (e), aumenta el número de operaciones a realizar para resolver la ecuación (1), con lo cual se requiere una alternativa para optimizar el proceso: La alternativa es emplear cadenas de adición para poder reducir el número de

multiplicaciones para resolver una potencia. Ejemplo: dado el exponente e= 91, en vez de multiplicar la base 91 veces es factible contar con una cadena asociada al número 91, como ser: $C = \{1, 2, 4, 6, 8, 14, 22, 36, 58, 80, 88, 90, 91\}$ y entonces resolver la potencia consistiría en multiplicar la base por cada uno de los exponentes de C, es decir que para este ejemplo se realizan sólo 13 operaciones.

Sin embargo y teniendo en cuenta que no existe una única cadena para un exponente, hallar una cadena de longitud mínima, constituye un problema de optimización combinatoria y puede abordarse empleando una metaheurística para evitar las pruebas por fuerza bruta para hallar un valor óptimo (cadena de longitud mínima).

1.1. Antecedentes

A continuación y a modo de síntesis, se presenta un resumen del estado del arte del problema objeto de estudio en el campo de las metaheurísticas. Se resume en el Cuadro 1 los resultados de estudios anteriores en donde se abordó el problema de optimización de la función objetivo definida por la ecuación (1) mediante el uso de distintas metaheurísticas ampliamente conocidas, ordenadas de forma cronológica por año e indicando los rangos del exponente «e» empleados para cada enfoque:

Cuadro 1. Resumen del estado del arte de la problemática objeto de estudio.

MH	ACO [7]	GA [8]	AIS [13]	PSO [14]	EP [17]	GA Híbrido [22]	GA [23]
Año	2004	2005	2008	2009	2011	2011	2016
$e \in$	Rangos de exponentes						
< 128	X	X	X	-	X	X	X
$[128, 512]$	X	-	X	-	X		
$[512, 1024]$	X	-	X	-			

Considerando que la presente propuesta se centra en el uso de la metaheurística GWO como estrategia de tratamiento para la problemática de generación de cadenas de adición, resulta interesante comparar con propuestas del estado del arte también basadas en metaheurísticas. Nedjah y De Macedo-Mourelle en 2004, experimentaron con ACO [7] para resolver el problema de hallar cadenas de longitud óptima, trabajando con exponentes de hasta una longitud de 128-bits de longitud, no abordando pruebas con exponentes de longitud mayor a dicho rango, hasta un trabajo posterior donde aplicaron ACO a exponentes de hasta 1024-bits. Cruz-Cortés y otros, en 2005, emplearon GA [8] logrando hallar cadenas óptimas para exponentes de todos los rangos menores a 4096 ($e < 4096$). En 2008 Rodríguez-Henríquez y Coello Coello, propusieron AIS [13], consiguiendo obtener cadenas mínimas para $e \in [1, 4096]$ pudiendo obtenerse

resultados para todos los rangos pero mejores para aquellos de longitud acotada entre [128,512], dejando propuesto como trabajo futuro lograr mejores resultados para exponentes grandes. León-Javier en 2009 implementó PSO [14] pero sólo reportó experimentos con exponentes pequeños, dejando los otros casos como trabajo pendiente. Isidro Domínguez en 2011 en su tesis adaptó el algoritmo EP [17] logrando minimizar cadenas para exponentes de los rangos 128, 256, 512 y 1024 bits. El mismo año se propuso una alternativa de hibridación de GA con "Simulated Annealing" [22] mejorando con esto la convergencia del AG, pero sin embargo los resultados no superan a otras propuestas anteriores como el caso de PSO con la cual se comparó su desempeño. Finalmente en 2016 se propuso un GA con una serie de mejoras en los operadores de variación, una representación novedosa de soluciones y una estrategia de reparación de soluciones [23]. Se optimizó exponentes en el rango $[2^{37}-3, 2^{127}-3]$.

2. Algoritmo basado en el comportamiento de lobos grises (Grey Wolf Optimizer)

En [21] se propone una metaheurística dominada Grey Wolf Optimizer (GWO) inspirada en el comportamiento de las manadas de lobos grises (Canis Lupus) y su organización social para la caza de presas. Los lobos grises por naturaleza prefieren vivir en grupo de entre 5 y 12 miembros en promedio. La manada se conforma de dos líderes llamados alfa (un macho y una hembra) y el resto de la manada. En esta estructura social, los miembros alfa son los responsables de la toma de decisiones sobre la caza, lugar donde dormir, hora de despertar, y el resto de los miembros debe seguir sus órdenes. El miembro alfa de una manada no necesariamente debe ser el más fuerte, pero sí el mejor en términos de conducción del grupo y capacidad para tomar decisiones. El segundo nivel en la jerarquía del grupo son los miembros beta (subordinados de los alfa), que deben colaborar con la toma de decisiones y el cumplimiento de las órdenes de los alfa. Estos miembros son los candidatos a convertirse en alfa cuando alguno de estos fallece o envejezca. Existe otra clasificación que se encuentra más abajo en la jerarquía de las manadas de lobos grises y se denomina omega, éstos deben someterse a los alfa y betas obedeciéndoles, en muchos casos suele verse en este tipo a miembros que ofician de niñeros de las crías, dentro de la manada [24]. Las principales fases de la caza en manadas de lobo gris son las siguientes:

- Seguir, perseguir, y acercarse a la presa.
- Perseguir, rodear, y acosar a la presa hasta que se "paraliza".
- Atacar a la presa.

Esta técnica de caza fue modelada matemáticamente en [21] con la finalidad de proponer un método de optimización y emplearlo en la resolución de problemas complejos.

2.1. Modelo matemático y algoritmo

A continuación se describe el modelo matemático de la jerarquía social de las manadas del lobo gris en base a las tres acciones principales relacionadas con la actividad de la caza: seguir, rodear (acorralar) y atacar.

Jerarquía social de las manadas:
Para modelar matemáticamente la jerarquía social de los lobos en el diseño de GWO, se considera que la solución más apta para el problema que se intenta resolver es la alfa (α). En consecuencia, la segunda y tercera mejores soluciones se denominan beta (β) y delta (δ), respectivamente. El resto de las soluciones candidatas, se supone que son omega (ω). En el algoritmo GWO la actividad de caza se traduce como proceso de optimización, y el mismo es guiado por α, β, y δ, mientras que los lobos ω siguen a los 3 tipos previamente descriptos.

Acorralamiento de la presa:
Los lobos grises rodean la presa durante la caza. Matemáticamente este comportamiento se modela de la siguiente manera:

$$D = |C.X_p(t) - X(t)| \tag{2a}$$
$$X(t+1) = X_p(t) - A \cdot D \tag{2b}$$

Donde t indica la iteración actual, A y C son vectores de coeficientes, X_p es el vector de posición de la presa, y X indica la posición vector de un determinado lobo gris. Por otro lado, los vectores A y C se calculan como sigue:

$$A = 2ar_1 - a \tag{3a}$$
$$C = 2r_2 \tag{3b}$$

Según la formulación original GWO en [21] los componentes del vector A disminuyen linealmente desde 2 hasta 0, sobre el curso de las iteraciones y r_1, r_2 son vectores aleatorios en el rango $[0, 1]$.

El modelo matemático de GWO describe la operación de acercamiento en función de que un lobo gris en una determinada posición (en el espacio n-dimensional), podrá actualizar su posición respecto de la presa, trasladando la ubicación del lobo (o agente) mediante el ajuste del valor de los vectores A y C.

Caza (ataque a la presa):
Los lobos grises tienen la capacidad de reconocer la ubicación de la presa y rodearla. En la naturaleza la caza suele ser guiada por el miembro alfa, aunque los miembros beta y delta que le siguen en jerarquía, pueden participar eventualmente de la caza. Con el fin de simular matemáticamente el comportamiento de caza de los lobos grises, suponemos que el miembro alfa (mejor solución candidata), beta y delta tienen un mejor conocimiento sobre la posible ubicación

de la presa. Por lo tanto, guardamos las primeras tres mejores soluciones obtenidas hasta el momento y obligamos a los otros agentes de búsqueda (incluidos los omegas) a actualizar sus posiciones de acuerdo a la posición de los mejores agentes de búsqueda. En este sentido se proponen las siguientes fórmulas:

$$D_\alpha = |C_1 X_\alpha - X|, D_\beta = |C_2 . X_\beta - X|, D_\delta = |C_3 X_\delta - X| \tag{4a}$$

$$X_1 = X_\alpha - A_1(D_\alpha), X_2 = X_\beta - A_2(D_\beta), X_3 = X_\delta - A_3(D_\delta) \tag{4b}$$

$$X(t+1) = (X_1 + X_2 + X_3)/3 \tag{4c}$$

En función de que cada agente actualiza su posición según: alfa, beta y delta en un espacio de búsqueda, se puede decir que el rol de alfa, beta y delta es estimar la posición de la presa y los otros lobos actualizan sus posiciones al azar alrededor de la misma.

Exploración (búsqueda) versus Explotación (ataque):

Los lobos se acercan a la presa y la atacan cuando la misma deja de moverse. El modelo matemático de tal comportamiento de acercarse a la presa, se traduce como la disminución del valor del vector A (coeficiente). Cabe destacar que el rango de fluctuación del vector A, también se reduce por el valor de "a" (3a). Cuando $|A| < 1$, GWO, obliga a los lobos a atacar a la presa. Esta característica está relacionada con la explotación del algoritmo.

GWO es propenso al estancamiento en las soluciones locales con estos operadores. Y si bien es cierto que el mecanismo de "rodeo" que posee genera exploración hasta cierto punto, necesita más operadores para enfatizar dicha exploración.

Los lobos tienen posiciones diferentes respectos de la presa pero convergen a la misma para atacar. El modelo matemático de esta divergencia se refleja en el uso de valores aleatorios para el vector A (mayores que 1 y menores que -1) para asegurar divergencia y que la búsqueda abarque de mejor manera el espacio de búsqueda, característica relacionada con la exploración, es decir que cuando $|A| > 1$ se está obligando a los lobos a divergir respecto de la presa.

En resumen, el proceso de búsqueda en GWO se inicia con la creación de población de lobos al azar (soluciones candidatas). En el transcurso de iteraciones, los lobos: alfa, beta y delta, estiman la posición probable de la presa. Cada solución candidata actualiza su distancia respecto de la presa. El parámetro "a" es disminuido desde 2 hasta 0 con el fin de enfatizar la exploración y explotación, respectivamente. Las soluciones candidatas divergen respecto de la presa cuando $|A| > 1$ y convergen hacia la presa cuando $|A| < 1$. Por último, el algoritmo GWO termina por la satisfacción de un criterio de fin. Tal como describe en [21] GWO ha sido aplicado a una serie de problemas de optimización y dentro de ellos se aplicó a un problema real del ámbito de la ingeniería óptica [21]. En el Algoritmo 1 se presenta el esquema general del método.

Algorithm 1 Pseudo código del algoritmo GWO.

Inicializar población de lobos X_i (con $i = 1, 2, ..., n$)
Inicializar α, A y C
Calcular fitness de cada agente de búsqueda
$X\alpha$ = el mejor agente.
$X\beta$ = el segundo mejor agente.
$X\delta$ = el tercer mejor agente.
while (t < número máximo de iteraciones) **do**
  for i = [1..n] **do**
    Actualizar la posición de cada agente mediante la ecuación (4c)
  end for
  Actualizar α, A y C
  Calcular el fitness de todos los agentes de búsqueda
  Actualizar X_α, X_β y X_δ
  t=t+1
end while
Retornar $X\alpha$

2.2. Propuesta

El objetivo del presente trabajo es abordar el problema de optimización de generación de cadenas de adición cuya longitud sea mínima y abordarlo con una metaheurística no explorada o es mente explorada para la problemática como lo es GWO y la cual es de más reciente surgimiento que las ya utilizadas para el problema de referencia. Se empleó una función programada sobre lenguaje MATLAB©, capaz de generar cadenas de adición asociada a exponentes que recibe como parámetros de entrada [17] (ver Algoritmo 2) y la misma fue optimizada mediante GWO.

Algorithm 2 Producir cadena de adición válida.

Asignar a $u_0 = 1$ y $u_1 = 2$
Seleccionar de manera aleatoria 3 o 4 y asignarlo a u_2.
Completar la secuencia invocando la función FILL(U, Rand(3,4),e) // parámetros que emplea la función para "llenar" la cadena.
Retornar (U, L)

En el Algoritmo 2, e es el exponente para el cual se genera la cadena de adición, $U : (u_0, u_1, u_2, \ldots, e)$ es la cadena de adición completa de longitud L, Rand(3,4) representa el método que toma de manera aleatoria el valor 3 o 4 para generar la tercera componente de la cadena y FILL es la función llenar, que se utiliza para completar la cadena y se describe en el Algoritmo 3.

Al igual que en [17] los individuos de la población serán generados aplicando las 3 reglas mencionadas a continuación, siempre con la premisa de que las dos primeras componentes de la cadena son 1 y 2, en este mismo orden:

Algorithm 3 FILL(U, Rand(3,4),e) // completar cadena.

```
Establecer i = k – 1
while U_i ≠ e // mientras la posición de la cadena sea distinta de e do
  if flip(f) then
    u_{i+1} = 2u_i // aplicar doblado (regla 1)
  else
    if FLIP(g) then
      u_{i+1} = u_i + u_{i−1} // aplicar (regla 2)
    else
      u_{i+1} = u_i + u_rand // aplicar (regla 3)
    end if
  end if
  while (u_{i+1}) > e // Para que la siguiente posición sea mayor que e do
    j = i – 1
    u_{i+1} = u_i + u_{i−j}
    j = j – 1 // se decrementa para no superar el valor de e.
  end while
end while
Retornar (U)
```

1. Se permite que un elemento de la cadena resulte de sumar al elemento previo por sí mismo. De este modo, en una cadena U, un componente ui+1 podrá ser igual a: $2u_i \equiv (u_{i+1} = u_i + u_i)$. Por ejemplo, para la cadena $1, 3, 6, 9, 12, 15, 21$, el tercer elemento se compone de sumar el segundo elemento consigo mismo.
2. Suma de dos números previos, de modo que: $u_{i+1} = u_i + u_{i-1}$. Tomando como ejemplo la misma cadena que en punto 1, el cuarto elemento (9) es igual al segundo sumado al tercero.
3. Suma de un número previo más un número aleatorio de la cadena, es decir: $u_{i+1} = u_i + u_{rand}$. Siguiendo el mismo ejemplo anterior: el elemento (15) se conforma de la suma del quinto (12) y el segundo elemento (3).

Para adaptar el comportamiento de GWO para resolver la problemática objeto de estudio, vamos a establecer que nuestra población de lobos (soluciones) es una población de cadenas de adición válidas para un exponente e determinado, es decir secuencias numéricas de enteros y que responden a las reglas mencionadas anteriormente. Por otro lado, teniendo en cuenta que el propósito de optimizar cadenas para un exponente determinado consiste en obtener cadenas de longitud mínima, la función de adaptación (fitness) a utilizar, será la longitud de las cadenas.

De esta manera, la función FILL aplica las reglas anteriores con determinados valores de probabilidad. Para los experimentos se ha tomado como referencia los valores reportados por el enfoque [17] que ha demostrado resultados competitivos para varios rangos de exponentes, a fin de poder comparar nuestra propuesta bajo condiciones similares en términos de generación de soluciones y desempeño de la metaheurística propuesta. Dichos valores son:

- Tasa de determinación de un elemento de la cadena a partir de la suma del número anterior por sí mismo (regla 1) = 0.7
- Tasa de suma de elementos anteriores (regla 2) = 0.2

En la siguiente sección, se presentan los resultados vertidos de una serie de pruebas estadísticas realizadas sobre la metaheurística GWO aplicada al problema objeto de estudio. Todos los resultados obtenidos con el enfoque GWO, han sido generados aplicando la siguiente configuración de parámetros:

- SearchAgents_no= 10 (número de agentes de búsqueda).
- Max_iteration= 30 (cantidad de ejecuciones).

3. Resultados obtenidos

En esta sección se describe de qué manera se probó el desempeño de GWO. Para todos los experimentos se utilizó la metaheurística GWO sobre MATLAB© v.7.8.0 (R2009a) bajo Sistema Operativo Windows 8.1 Enterprise 64 bits, en un computador: Intel Core i-3. 2.10 GHz., 4 Gb. RAM.

3.1. Experimento 1

El primer experimento planteado radica en calcular el total de longitudes acumuladas de cadenas aditivas para conjuntos de exponentes de pequeña longitud, a fin de comparar el desempeño de algoritmo GWO con otros resultados obtenidos por métodos deterministas de la literatura y por otras metaheurísticas del estado del arte.

Cuadro 2. Promedio de resultados obtenidos por GWO sobre los siguientes rangos de exponentes con 30 ejecuciones independientes.

Exponente	Longitud acumulada	Promedio	Mediana	Desviación
[1, 128]	960	7,5	4	2,159
[1, 256]	2459	9,6	9	2,986
[1, 512]	4994	9,8	10	1,9501

El Cuadro 2 resume los mejores resultados obtenidos por enumeración para los exponentes de los rangos expresados, tomando el mejor de un lote de 30 ejecuciones independientes, a fin de poder comparar tales resultados con otras propuestas para las que se han ejecutado experimentos empleando el mismo rango de exponentes.

El Cuadro 3 muestra una comparación de los resultados para GWO respecto de los métodos deterministas del estado del arte tomados de [13] cuyos resultados fueron utilizados para comparar con el enfoque AIS [13]. Por otro lado, el Cuadro 3 brinda una comparación de los mejores resultados de cadenas de

Cuadro 3. Cadenas de adición acumuladas para todas las longitudes de exponentes $e \in [1, 512]$

e	$[1, 512]$
Optimal [13]	4924
Binary [13]	5388
Çuaternary [13]	5226
Resultados GWO	
Longitudes acumuladas	4994
Promedio	9,77
Mediana	10
Desvío	1,95

Cuadro 4. Comparación de los mejores resultados acumulados para GWO con otros enfoques, para el mismo rango de exponentes.

$e \in [1, 512]$	AIS	GA	PSO	EP	GWO
	4924	4924	—	4924	4994

adición acumuladas para el mismo rango de exponentes $[1, 512]$ obtenidos por métodos estocásticos y comparados en [17], salvo PSO que no reportó resultados para este rango de exponentes, respecto de la propuesta GWO. En el Cuadro 3 se puede observar además que GWO se aproxima al resultado del método "Optimal"[13], siendo éste quien mejor desempeño logró para dicho rango de exponentes según [13]. Finalmente, en el Cuadro 4 se observa que GWO llega en su mejor ejecución igualar los mejores resultados que arrojaron AIS, GA, PSO, EP para el rango [1, 512]. En comparación con los resultados de los métodos deterministas se observa que en su mejor ejecución se aproxima al mejor resultado arrojado por "Optimal", mientras que en los resultados comparativos con los demás métodos estocásticos se acerca al óptimo de las otras propuestas, no superando las mismas por un porcentaje de $\sim 0,9\,\%$.

3.2. Experimento 2

El experimento plantea la comparación del desempeño de GWO respecto de una propuesta reciente [22] del estado del arte.

El Cuadro 5 muestra una comparación de los resultados de GWO respecto de GA Annealing [22] para una serie de exponentes diversos y especialmente difíciles de optimizar. Se considera dentro de esta categoría a aquellos exponentes para los cuales no es posible hallar cadenas de longitud mínima mediante los métodos deterministas. Se puede observar que en los 5 casos, GWO ha obtenido cadenas de longitud igual o menor al enfoque propuesto en [22]. Si bien los métodos aplicados en [13], [14] y [17] también han sido probados con esta clase de exponentes diversos, no son los mismos que los empleados en [22]. Por esta

Cuadro 5. Comparación de GA Annealing [22] y GWO para el mismo conjunto de exponentes.

Exponente	Cadena	GA Annealing	GWO
e		Longitud	
23	1, 2, 4, 5, 10, 20, 21, 23	7	7
55	1, 2, 3, 6, 12, 24, 27, 54, 55	9	8
130	1, 2, 4, 8, 16, 32, 64, 128, 129, 130	11	9
250	1, 2, 3, 5, 10, 20, 30, 50, 100, 150, 250	13	10
768	1, 2, 3, 6, 12, 24, 48, 96, 192, 384, 768	23	10

razón, se efectúa la comparación en el Cuadro 5 y en el próximo experimento se hace respecto a los otros métodos del estado del arte.

3.3. Experimento 3

El experimento plantea la comparación del desempeño de GWO respecto de otros abordajes del estado del arte.

Cuadro 6. Mejores resultados obtenidos por AIS [13], PSO [14], EP [17] y GWO para un conjunto de exponentes diversos.

Exponente		Longitud			
e	Cadena	AIS [13]	PSO [14]	EP [17]	GWO
5	1, 2, 3, 5	3	3	3	3
7	1, 2, 4, 5, 7	4	4	4	4
11	1, 2, 4, 8, 10, 11	5	5	5	5
19	1, 2 ,4, 8, 16, 18, 19	6	6	6	6
29	1, 2, 3, 6, 7, 14, 28, 29	7	7	7	7
47	1, 2, 4, 8, 9, 18, 36, 45, 47	8	8	8	8
71	1, 2, 3, 6, 7, 14, 21, 35, 70, 71	9	9	9	9
127	1,2, 3, 6, 12, 18, 30, 60, 63, 126, 127	10	10	10	10
191	1, 2, 3, 6, 12, 18, 19, 38, 57, 95, 190, 191	11	11	11	11
379	1, 2, 3, 6, 9, 18, 27, 45, 72, 117, 189, 378, 379	12	12	12	12
607	1, 2, 3, 6, 9, 15, 30, 60, 61, 121, 182, 303, 606, 607	13	13	13	13
1087	1, 2, 3, 6, 9, 18, 36, 54, 108, 216, 324, 540, 541, 1082, 1085, 1087	14	14	14	15

El Cuadro 6 resume los resultados del tercer experimento, correspondiente a los mejores resultados obtenidos por GWO para un conjunto de exponentes diversos y difíciles de optimizar al igual que los resultados del experimento mostrados en el Cuadro 5 pero comparando dichos resultados con los obtenidos por

AIS [13], PSO [14] y EP [17] para los mismos casos. GWO ha sido comparado tomando algunos exponentes de las metaheurísticas mencionadas en el párrafo anterior a excepción de [8] pues tal como se menciona en [17], todas las demás ([13], [14]) superan sus resultados. Si bien GWO no supera los resultados obtenidos para los exponentes listados mediante el uso de los otros enfoques (ver Cuadro 6) tampoco ha arrojado resultados de menor calidad en términos de longitud de cadena, salvo para el caso del exponente e = 1083 empleado en este lote de casos de prueba. Al margen de coincidir la mayoría de los casos tomados para el experimento con los resultados arrojados por otras metaheurísticas, dichos resultados corresponden a cadenas de longitud mínima en función del cuadro de exponentes "diversos" presentado en [13]:

Cuadro 7. Mejores resultados obtenidos por AIS [13], PSO [14], EP [17] y GWO para un conjunto de exponentes diversos.

Longitud de cadena	Exponentes (e)
1	2
2	3, 4
3	5, 6, 8
4	7, 9, 10, 12, 16
5	11, 13, 14, 15, 17, 18, 20, 24, 32
6	19, 21, 22, 23, 25, 26, 27, 28, 30, 33, 34, 36, 40, 48, 64
7	29, 31, 35, 37, 38, 39, 41, 42, 43, 44, 45, 46, 49, 50, 51, 52, 54, 56, 60, 65, 66, 68, 72, 80, 96, 128
8	47, 53, 55, 57, 58, 59, 61, 62, 63, 67, 69, 70, 73, 74, 75, 76, 77, 78, 81, 82, 83, 84, 85, 86, 88, 90, 92, 97, 98, 99, 100, 102, 104, 108, 112, 120, 129, 130, 132, 136, 144, 160, 192, 256
9	71, 79, 87, 89, 91, 93, 94, 95, 101, 103, 105, 106, 107, 109, 110, 111, 113, 114, 115, 116, 117, 118, 119, 121, 122, 123, 124, 125, 126, 131, 133, 134, 135, 137, 138, 140, 145, 146, 147, 148, 149, 150, 152, 153, 154, 156, 161, 162, 163, 164, 165, 166, 168, 170, 172, 176, 180, 184, 193, 194, 195, 196, 198, 200, 204, 208, 216, 224, 240, 257, 258, 260, 264, 272, 288, 320, 384, 512

El Cuadro 7 resume los casos de exponentes más conocidos y probados de la literatura para los que se conoce cadenas de longitud óptima, indicando a qué longitud corresponden dichos valores óptimos (primera columna del cuadro). En función de esto, se verifica que por ejemplo el exponente e = 71 que se utilizó para el experimento del Cuadro 6, para el cual se obtuvo la cadena 1, 2, 3, 6, 7, 14, 21, 35, 70, 71 cuya longitud es 9, ésta coindice con la longitud que indica el Cuadro 7, además de existir coincidencia en la longitud también obtenida por [13], [14] y [17] para el mismo caso.

4. Conclusión

En este trabajo se propone el uso de la heurística GWO para encontrar cadenas de adición de longitud mínima, asociadas a distintos exponentes. GWO demostró resultados competitivos para abordar la problemática, que se evidencia a partir de la comparación de sus resultados, respecto de otras propuestas del estado arte. En el Experimento 1 se observa que para el rango de exponentes [1, 512], los resultados de cadenas de adición mínimas acumuladas superan ampliamente a dos de los métodos deterministas reportados en [13] para dicho rango y sólo es superado escasamente por "Optimal" [13]. Respecto de las propuestas de abordaje con métodos estocásticos, en el segundo experimento ha sido comparado de forma directa con los mismos casos de exponentes reportados por [22], logrando igualar y superar dicha propuesta, en la mayoría de los casos. Por último, se ha comparado la propuesta para un grupo de exponentes diversos y particularmente difíciles de optimizar, con relación al desempeño de otras metaheurísticas de la literatura: AIS [13], PSO [14] y EP [17], logrando igualar a aquellos los resultados reportados a excepción del exponente de mayor tamaño.

Como trabajo futuro, se analizarán los parámetros requeridos por GWO [21] o incluso hibridaciones u otras técnicas requeridas, para ampliar los resultados de la metaheurística a exponentes de mayor longitud.

Referencias

1. Rivest. R.L.; Shamir, A.; Adlman, L. 1978. *A method for obtaining digital signatures and public-key cryptosystems.* Cambridge.
2. Bos, J.; Coster, M. 1990. *Addition Chain Heuristics. Centrum voorWiskunde en Infomatica.* Amsterdam, Netherland.
3. Kaya Koc, C. 1994. *High-speed RSA implementation.* Technical report, RSA. Laboratories, Redwood City, CA. USA.
4. Rotger, L.H.; Coma, J.R.; Tena-Ayuso, J.G. *Criptografía con Curvas Elípticas.* Universitat Oberta de Catalunya. España
5. Liu, Y.; Passino, K.M. 2002. *Biomimicry of Social Foraging Bacteria for Distributed Optimization: Models, Principles, and Emergent Behaviors.* Journal of Optimization Theory and Applications: Vol. 115, No. 3, pp. 603–628. Ohio State University, Columbus, Ohio.
6. Nicosia, G.; Cutello, V.; Bentley, P.; Timmis, J. 2004. *Artificial Immune Sytems.* Third International Conference, ICARIS 2004. Springer- Verlag Berlin Heidelberg, Germany.
7. Nedjah, N.; De Macedo-Mourelle, L. 2004. *Finding Minimal Addition Chains Using Ant Colony.* Department of Systems Engineering and Computation Faculty of Engineering, State University of Rio de Janeiro. Río de Janeiro, Brasil.
8. Cruz-Cortés, N; Rodríguez-Henríquez, F.; Juárez-Morales, R.; Coello Coello, C. 2005. *Finding Optimal Addition Chains Using a Genetic Algorithm Approach.* México.
9. Karaboga, D. 2005. *An Idea Based on Honey Bee Swarm for Numerical Optimization.* (Technical Report). Erciyes University, Engineering Faculty Computer Engineering Department. Kayseri, Turquía.

10. Karaboga, D.; Basturk, B. 2007. *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm.* Department of Computer Engineering, Erciyes University, Kayseri, Turkey.
11. Washington, L. 2008. *Elliptic Curves: Number Theory and Cryptography.* Second Edition. Discrete Matemathics and Its applications, series editor: Kenneth H. Rosen. Taylor & Francis Group. Boca Ratón, USA.
12. Mezura-Montes, E.; Hernández- Ocaña, B. 2008. *Bacterial Foraging for Engineering Design Problems: Preliminary Results.* Laboratorio Nacional de Informática Avanzada (LANIA A.C.) - Universidad Juárez Autónoma de Tabasco. México.
13. Cruz-Cortés, N.; Rodríguez-Henríquez, F.; Coello Coello, C. 2008. *An Artificial Immune System Heuristic for Generating Short Addition Chains.* México.
14. León. A.; Cruz-Cortés, N.; Moreno-Armendáriz, M.; Orantes-Jiménez, S. 2009. *Finding Minimal Addition Chains with a Particle Swarm Optimization Algorithm.* Center for Computing Research, National Polytechnic Institute. México.
15. Yang, X.S. 2010. *Nature – Inspired Metaheuristic Algorithms*, Second Edition. University of Cambridge. Luniver Press, United Kingdom.
16. Gómez Bello, M. 2011. *La aritmética modular y alguna de sus aplicaciones.* Universidad Nacional de Colombia. Bogotá, Colombia.
17. Domínguez, I. 2011. *Optimización de Cadenas de Adición en Criptografía utilizando Programación Evolutiva.* Tesis de Maestría. Laboratorio Nacional de Informática Avanzada Centro de Enseñanza LANIA, Xalapa, Veracruz, México.
18. Binitha, S.; S Siva Sathya. 2012. *A Survey of Bio inspired Optimization Algorithms.* International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2. India.
19. Baro, M. 2013. *Swarming: La Comunicación en múltiples direcciones y múltiples etapas.* Razón y palabra. Primera Revista Electrónica en Iberoamérica Especializada en Comunicación. Centro Avanzado de Comunicación - 25 Aniversario Eulalio Ferrer. Número 83, Junio – Agosto
20. Tall, A.; Sanghare, A.Y. 2013. *Efficient computation of addition-subtraction chains using generalized continued Fractions.* African Institute for Mathematical Sciences. Senegal.
21. Mirjalili,S.; Mirjalili, S.M.; Lewis, A. 2014. *Grey Wolf Optimizer.* School of Information and Communication Technology, Griffith University, Nathan Campus, Brisbane QLD 4111, Australia. Department of Electrical Engineering, Faculty of Electrical and Computer Engineering, ShahidBeheshti University, G.C. 1983963113, Tehran, Iran.
22. Pogančić, M. 2014. *Evolving Minimal Addition Chain Exponentiation.* University of Zagreb - Faculty of Electrical Engineering and Computing. Zagreb, Croacia.
23. Picek, S.; Coello Coello, A.; Jakobovic, D.; Metens, N. 2016. *Evolutionary Algorithms for Finding Short Addition Chains: Going the Distance.* Volume 9595 of the series Lecture Notes in Computer Science pp 121-137.
24. Mech., D. 1999. *Alpha Status, Dominance, and Division of Labor in Wolf Packs.* Canadian Journal of Zoology. Jamestown, Dakota del Norte.